(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Veröffentlichungsnummer: **0 450 116 A1**

(12) EUROPÄISCHE PATENTANMELDUNG

(21) Anmeldenummer: **90106297.6**

(51) Int. Cl.5: **G06F 11/00**

(22) Anmeldetag: **02.04.90**

(43) Veröffentlichungstag der Anmeldung:
09.10.91 Patentblatt 91/41

(84) Benannte Vertragsstaaten:
CH DE FR IT LI

(71) Anmelder: **SIEMENS AKTIENGESELLSCHAFT**
Wittelsbacherplatz 2
W-8000 München 2(DE)

(72) Erfinder: **Trummer, Georg, Dipl.-Ing. (FH)**
Jahnstrasse 1
W-8453 Vilseck(DE)

(54) Automatisierungsgerät mit Test in einzelnen Schritten.

(57) Bei Automatisierungsgeräten (1) mit einem Standardprozessor (3) und mit einem Compiler (8) übersetzt letzterer zur schnellen Abarbeitung ein im Maschinencode vorliegendes Anwenderprogramm in die Maschinensprache des Standardprozessors (3). Der Standardprozessor (3) arbeitet das Anwenderprogramm ab. Zum Programmtest wurde bisher der Maschinencode interpretativ vom Standardprozessor mit entsprechendem Zeitaufwand abgearbeitet. Der viel schnellere befehlsgranulare Oneline-Test im Maschinencode ist möglich, indem ein Speicher vorgesehen wird zur Abspeicherung einer Testinformation, die für den zu testenden Bereich des Anwenderprogramms die Zuordnung zwischen den Adressen jedes Befehls im Maschinencode zu den Adressen der entsprechenden Befehle in der Maschinensprache des Standardprozessors (3) beinhaltet. Weiterhin sind Mittel (5) zur Unterbrechung des Programmablaufs vorgesehen.

EP 0 450 116 A1

Die Erfindung betrifft ein Automatisierungsgerät mit einem ersten Speicher zur Speicherung eines Anwenderprogramms im Maschinencode mit einem Standardprozessor, der in seiner eigenen Maschinensprache arbeitet, mit einem Compiler, der zur Übersetzung des im Maschinencode stehenden Anwenderprogramms in die Maschinensprache des Standardprozessors dient, in der das Anwenderprogramm für den Standardprozessor abarbeitbar ist.

Automatisierungsgeräte der obengenannten Art sind bekannt. Eine wichtige Anforderung an Automatisierungsgeräte ist die Möglichkeit sie Online, d.h. während des Betriebs testen zu können. Bisher bekannte Testfunktionen basieren auf der Möglichkeit, das im Maschinencode vorliegende Anwenderprogramm im Einzelschrittbetrieb ablaufenzulassen. Hierbei wird nach Ausführung jedes Befehls die Abarbeitung des Anwenderprogramms unterbrochen, um die Testdaten sammeln zu können. Bei Automatisierungsgeräten der obengenannten Art ist ein solcher Online-Test nicht anwendbar, weil hierbei das Anwenderprogramm in der Maschinensprache des Standardprozessors abgearbeitet wird. Hierbei stellt sich das Problem, daß der Maschinencode des Automatisierungsgeräts zur Maschinensprache in seinen Befehlsgrenzen nicht proportional ist. Demzufolge ist nicht bekannt, welche Unterbrechungsstellen in der Maschinensprache den Befehlsgrenzen des Maschinencodes entsprechen. Die beschriebene Einzelschrittbearbeitung zum Online-Test ist daher bei Abarbeitung des Anwenderprogramms in der Maschinensprache nicht möglich. Um dennoch den Maschinencode im Einzelschrittbetrieb testen zu können, wurde bisher der Test in einer anderen Betriebsart vorgenommen, in der der Maschinencode interpretativ vom Standardprozessor abgearbeitet wird. Diese interpretative Abarbeitung geschieht in Software und benötigt eine erhebliche Bearbeitungszeit pro Befehl, die sich wie etwa folgt aufteilt:

1. Opcode Fetch (5 m/sec.)
2. Decodieren (35 m/sec.)
3. Ausführen (30 m/sec.)

Dies bedeutet, daß für Online-Tests erhebliche Bearbeitungszeiten benötigt werden. Der an sich durch Verwendung des Compilers gegebene Vorteil, das Anwenderprogramm in der Maschinensprache des Standardprozessors schnell abarbeiten zu können, kommt hier überhaupt nicht zum Tragen. Es besteht daher die Aufgabe, ein Automatisierungsgerät der obengenannten Art zu schaffen, das einen Online-Test befehlsgranular im Maschinencode ermöglicht, wobei die Abarbeitung des Anwenderprogramms in der Maschinensprache des Standardprozessors erfolgt.

Dies wird dadurch gelöst, daß ein zweiter Speicher vorgesehen ist, zur Abspeicherung einer Testinformation, die für den zu testenden Bereich des Anwenderprogramms die Zuordnung zwischen den Adressen jedes Befehls im Maschinencode zu den Adressen der entsprechenden Befehle in der Maschinensprache des Standardprozessors beinhaltet und daß Mittel vorgesehen sind, die zur Sammlung von Testdaten an einer vorwählbaren Befehlsadresse im Maschinencode die Abarbeitung des Anwenderprogramms durch den Standardprozessor unterbrechen, wenn die dieser Befehlsadresse entsprechenden Befehlsadressen in der Maschinensprache gemäß der Testinformation bei der Abarbeitung des Anwenderprogramms in der Maschinensprache erreicht sind. Es erweist sich als vorteilhaft, wenn die Mittel als Adressvergleicher ausgeführt sind, da dieser leicht realisierbar ist. Es ist jedoch auch ein Vorteil, wenn ein dritter Speicher vorgesehen ist, in dem ein Unterbrechungsprogramm abgespeichert ist, mit welchem nach Ausführung des Befehls eine Unterbrechung der Abarbeitung des Anwenderprogramms zur Sammlung von Testdaten erfolgt. Diese Weiterbildung erlaubt, den Prozessor unabhängig von seiner Prozessorarchitektur stets nach jedem Befehl anzuhalten. Eine besonders anwenderfreundliche und vorteilhafte Ausführung ist dadurch gegeben, wenn das Anwenderprogramm in einem bestimmten Maschinencode über ein Programmiergerät in das Automatisierungsgerät eingegeben wird, wobei das Programmiergerät selbst die Umsetzung eines in einer anwenderfreundlichen Programmiersprache eingegebenen Anwenderprogramms in den Maschinencode vornimmt. Diese Ausführung schafft die Kompatibilität zwischen verschiedenen Programmier- und Automatisierungsgeräten. Zur Durchführung des Online-Tests eines Anwenderprogramms bei einem Automatisierungsgerät eignet sich besonders gut das Verfahren nach Anspruch 5. Dieses ermöglicht, den Online-Test unter Beibehaltung der schnellen Abarbeitung des Anwenderprogramms in der Maschinensprache vorzunehmen.

Im folgenden wird ein Ausführungsbeispiel der Erfindung anhand einer Zeichnung näher erläutert. Es zeigen:

FIG 1 die Hardware-Architektur eines Automatisierungsgerätes mit daran angeschlossenem Programmiergerät,

FIG 2 die Zuordnung der Befehlsgrenzen im Maschinencode zu denen in der Maschinensprache des Prozessors mit Hilfe einer Liste,

FIG 3 den Ablaufplan für einen Test.

FIG 1 zeigt ein Automatisierungsgerät 1 mit einem ersten Speicher 2, einem zweiten Speicher 4, einem dritten Speicher 6, einem Adressvergleicher 5 und einem Compiler 8. Mit dem Automatisierungsgerät 1 ist ein Programmiergerät 7 verbunden, über das kundenseitig ein Anwenderprogramm eingegeben werden kann. Zur Eingabe die-

ses Anwenderprogramms stehen z.B. drei Alternativen zur Verfügung:

1. eine Anweisungsliste AWL,
2. ein Koppelplan KOP und
3. ein Funktionsplan FUP (siehe Siemens Steuerungen programmieren mit Step 5, Berger, Band 2, 1983).

Das Programmiergerät 7 generiert aus dieser Eingabeinformation einen bestimmten Maschinencode, in dem das Anwenderprogramm dem Automatisierungsgerät 1 eingegeben wird. Bei Verwendung dieses stets gleichen Maschinencodes für Automatisierungsgeräte verschiedener Typen und auch bei verschiedenen Programmiergeräten 7 sind diese untereinander kompatibel. Mit Hilfe des Compilers 8 wird das Anwenderprogramm vom Maschinencode in die Maschinensprache des Standardprozessors 3 übersetzt. Dem Automatisierungsgerät 1 wird, z.B. über das Programmiergerät 7, vorgegeben, nach welchen ausgeführten Befehlen im Maschinencode bei der Abarbeitung des Anwenderprogramms Testdaten gesammelt werden sollen. Da aber das Anwenderprogramm in der Maschinensprache des Standardprozessors 3 abgearbeitet wird, müssen die entsprechenden Unterbrechungsstellen des Anwenderprogramms in der Maschinensprache bestimmt werden, die den gewählten Unterbrechungsstellen im Maschinencode entsprechen. Diese als Testinformation bezeichnete Zuordnung der jeweiligen Befehlsadressen wird in einer Liste gemäß FIG 2 eingetragen. Da die Online-Tests meist nur für jeweils bestimmte Programmbausteine innerhalb des Anwenderprogramms durchgeführt werden, ist es vorteilhaft, die Testinformation jeweils auf diesen Programmbaustein zu beschränken. In diesem Fall ist der zur Abspeicherung der Testinformation erforderliche Speicherbedarf entsprechend gering. Der Testinformation ist nur temporär in einem hierfür vorgesehenen Speicher abgespeichert, denn der Speicherinhalt wird bei einem nachfolgenden Test eines anderen Programmbausteins mit dessen zugehöriger Testinformation überschrieben. Die Testinformation ist eine Liste, in der die Positionen der Listeneinträge proportional zum Maschinencode sind und die Listeneinträge auf die relevanten Stellen in der Maschinensprache zeigen.

FIG 2 zeigt, daß je Maschinenbefehl mehrere Unterbrechungsursachen denkbar sind, um die Testdaten sammeln zu können. In FIG 2 sind zwei Unterbrechungsursachen je Maschinenbefehl realisierbar. Aus der Zuordnung zwischen den Adressen im Maschinencode und den Adressen in der Maschinensprache des Prozessors, die mit der Testinformation gegeben sind, ist es möglich, Testdaten nach Ausführung bestimmter Befehle im Maschinencode zu sammeln. Dabei ist zu beachten, daß die Adressen, auf den betreffenden Programmbaustein bezogen, relative Adressen sein können, zu denen die entsprechenden absoluten Adressen im Anwenderprogramm unter Berücksichtigung des betrachteten Programmbausteins bestimmt werden. Zur Abspeicherung der Testinformation dient der zweite Speicher 4. Um gemäß der Testinformation das Anwenderprogramm bei der Abarbeitung zur Sammlung von Testdaten zu unterbrechen, wird der Adressvergleicher 5 eingesetzt, der die aktuelle Adresse bei der Abarbeitung des Anwenderprogramms mit der jeweiligen Adresse vergleicht, bei der eine Unterbrechung erfolgen soll. Bei Übereinstimmung beider Adressen gibt der Adressvergleicher 5 ein Unterbrechungssignal an den Standardprozessor 3. Aufgrund dieser Unterbrechung werden die anstehenden Testdaten gesammelt. Vorteilhafterweise wird die Maschinensprache des Prozessors, d.h. der zu testende Code bei diesem Testverfahren überhaupt nicht geändert. Weitere Vorteile sind, daß der Test nur einen geringen zeitlichen Offset mit sich bringt und der Test im Originalcode, d.h. dem Maschinencode durchgeführt wird. Außerdem kann das Anwenderprogramm an Stellen angehalten werden, die nicht in einem schreibbaren Speicherbereich liegen. Allerdings muß der Prozessor befehlsgranular unterbrechbar sein.

Alternativ zu dem hardwaremäßig ausgeführten Adressvergleicher 5 ist es auch denkbar, einen speziellen Programmteil vorzusehen, der in dem dritten Speicher 6 abgelegt ist. Mit diesem als Software-Anker zu betrachtenden Programmteil wird das in der Maschinensprache stehende Anwenderprogramm, das der Unterbrechungsstelle folgt, überschrieben. Solange an der Stelle die Testdaten gesammelt werden, unterbricht dieser Programmteil die Abarbeitung des Anwenderprogramms. Zur Fortsetzung des Anwenderprogramms wird dieser Programmteil wieder mit den ursprünglich an dieser Stelle gestandenen Befehlen des Anwenderprogramms überschrieben. Diese Lösung weist den Vorteil auf, daß der Standardprozessor unabhängig von seiner Prozessorarchitektur stets angehalten werden kann. Dies ist bei der zuvor beschriebenen Hardware-Lösung mit dem Adressvergleicher 5 nicht immer definiert möglich. Außerdem besteht auch hier der Vorteil eines nur geringen zeitlichen Offsets durch den Test. Nachteilig ist jedoch bei der Ausführung mit dem Software-Anker, daß das Anwenderprogramm nur dort angehalten werden kann, wo der Speicher beschreibbar ist. Weiterhin wird beim Test nicht der Originalcode abgearbeitet.

Bild 3 zeigt den Ablauf des Online-Tests in einem Ablaufplan (AG-Protokoll). Der Online-Test wird dadurch eingeleitet, daß über das Programmiergerät 7 ein Testauftrag an die Zentraleinheit des Prozessors 3 mit Informationen über den zu

testenden Baustein und die relativen Adressen der Einzelschrittbearbeitung, bezogen auf den Bausteinbeginn, gegeben werden. In der Zentraleinheit ist nun über die Testinformationsliste die absolute Adresse der Unterbrechung im Code der Maschinensprache abfragbar. Die absolute Adresse der Unterbrechungsstelle wird in einem hardwaremäßig vorhandenen Adressvergleicher 5 geladen, der den Adressbus bezüglich der zu den Befehlen gehörenden Opcode-Fetch-Zugriffe des Prozessors 3 auf die angegebene Adresse überwacht. Erkennt der Adressvergleicher 5 diesen Zugriff, meldet er dies dem Prozessor durch ein Unterbrechungssignal (siehe FIG 2). Der Prozessor 3, der selbst den Zugriff durchgeführt hat, wird nach Bearbeitung des zuvor geholten Befehlscodes durch den Adressvergleicher 5 unterbrochen.

In der Interrupt-Service-Routine werden die Testdaten gesammelt. Danach wird anhand des Testauftrags und der Testinformation entschieden, ob weitere Unterbrechungen erforderlich sind.

Falls weitere Unterbrechungen notwendig sind, wird in der Interrupt-Service-Routine der Adressvergleicher 5 neu geladen, ansonsten bleibt er unangetastet. Ist die Interupt-Service-Routine beendet, so wird die Programmbearbeitung an der Unterbrechungsstelle fortgesetzt. Dieser Mechanismus wiederholt sich bis zur Deaktivierung des Adressvergleichers 5. Der Online-Test nach dem zuvor beschriebenen Verfahren ist nur durchzuführen, wenn zwei Anforderungen erfüllt sind:

1. Der Prozessor muß befehlsgranular unterbrechbar sein und
2. die betreffende Unterbrechung darf nicht maskiert sein, bzw. muß im Verhältnis zur Abarbeitung des Codes in der Maschinensprache die richtige Prioritätslage haben.

Die Forderung nach der Nichtmaskierung bedeutet, daß der Prozessor imstande sein muß, den Interrupt des Adreßvergleichers 5 zu bemerken. Hierzu darf der Interruptkanal im Prozessor nicht gesperrt sein.

Im Prozessor haben Interrupts bestimmte Prioritäten. Die Interrupt-Service-Routine hat immer die gleiche Priorität wie der Interrupt selbst. Interrupt-Service-Routinen mit niederer Priorität können durch höherpriore Interrupts mit den daran hängenden Interrupt-Service-Routinen unterbrochen werden. Niederpriore Interrupts können höherpriore Interrupt-Service-Routinen nicht unterbrechen. Beim vorliegenden Test ist die Programmbearbeitung niederprior als der Interrupt durch den Adreßvergleicher 5. Folglich können Testdaten gesammelt werden, da der Prozessor den Interrupt vom Andreßvergleicher 5 durchläßt, solange das niederpriore Programm läuft.

Patentansprüche

1. Automatisierungsgerät (1) mit einem ersten Speicher (2) zur Speicherung eines Anwenderprogramms im Maschinencode, mit einem Standardprozessor (3), der in seiner eigenen Maschinensprache arbeitet, mit einem Compiler (8), der zur Übersetzung des im Maschinencode stehenden Anwenderprogramms in die Maschinensprache des Standardprozessors (3) dient, in der das Anwenderprogramm für den Standardprozessor (3) abarbeitbar ist, **dadurch gekennzeichnet,** daß ein zweiter Speicher (4) vorgesehen ist, zur Abspeicherung einer Testinformation, die für den zu testenden Bereich des Anwenderprogramms die Zuordnung zwischen den Adressen jedes Befehls im Maschinencode zu den Adressen der entsprechenden Befehle in der Maschinensprache des Standardprozessors beinhaltet, und daß Mittel (5) vorgesehen sind, die zur Sammlung von Testdaten an einer vorwählbaren Befehlsadresse im Maschinencode die Abarbeitung des Anwenderprogramms durch den Standardprozessor (3) unterbrechen, wenn die dieser Befehlsadresse entsprechenden Befehlsadressen in der Maschinensprache gemäß der Testinformation bei der Abarbeitung des Anwenderprogramms in der Maschinensprache erreicht sind.

2. Automatisierungsgerät nach Anspruch 1, **dadurch gekennzeichnet,** daß die Mittel als Adressvergleicher (5) ausgeführt sind.

3. Automatisierungsgerät nach Anspruch 1, **dadurch gekennzeichnet,** daß ein dritter Speicher (6) vorgesehen ist, in dem ein Unterbrechungsprogramm abgespeichert ist, mit welchem nach Ausführung des Befehls eine Unterbrechung der Abarbeitung des Anwenderprogramms zur Sammlung von Testdaten erfolgt.

4. Automatisierungsgerät nach einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet,** daß das Anwenderprogramm in einem bestimmten Maschinencode über ein Programmiergerät (7) in das Automatisierungsgerät (1) eingegeben wird, wobei das Programmiergerät (7) selbst die Umsetzung eines in einer anwenderfreundlichen Programmiersprache eingegebenen Anwenderprogramms in den Maschinencode vornimmt.

5. Verfahren zum Online-Test eines Anwenderprogramms bei einem Automatisierungsgerät nach Anspruch 1, **dadurch gekennzeichnet,** daß in einem ersten Schritt dem Automatisie-

rungsgerät (1) die Adressen der Befehle im zu testenden Bereich des Anwenderprogramms im Maschinencode eingegeben werden, nach deren Ausführung in der Maschinensprache Testdaten gesammelt werden, im zweiten Schritt ein Übersetzungslauf des in Maschinencode vorliegenden Anwenderprogramms in die Maschinensprache des Standardprozessors (3) erfolgt, bei dem für den zu testenden Bereich des Anwenderprogramms die Zuordnung zwischen den Adressen jedes Befehls im Maschinencode zu den Adressen der entsprechenden Befehle in der Maschinensprache des Standardprozessors (3) bestimmt wird, im dritten Schritt als Testinformation die Zuordnung in dem zweiten Speicher (4) abgelegt wird, im vierten Schritt der Adressvergleicher (5) die Abarbeitung des Anwenderprogramms durch den Standardprozessor unterbricht, wenn die den im ersten Schritt eingegebenen Adressen entsprechenden Befehlsadressen in der Maschinensprache gemäß der Testinformation bei der Abarbeitung des Anwenderprogramms in der Maschinensprache erreicht werden und im fünften Schritt die bei jeder Unterbrechung anstehenden Testdaten gesammelt werden.

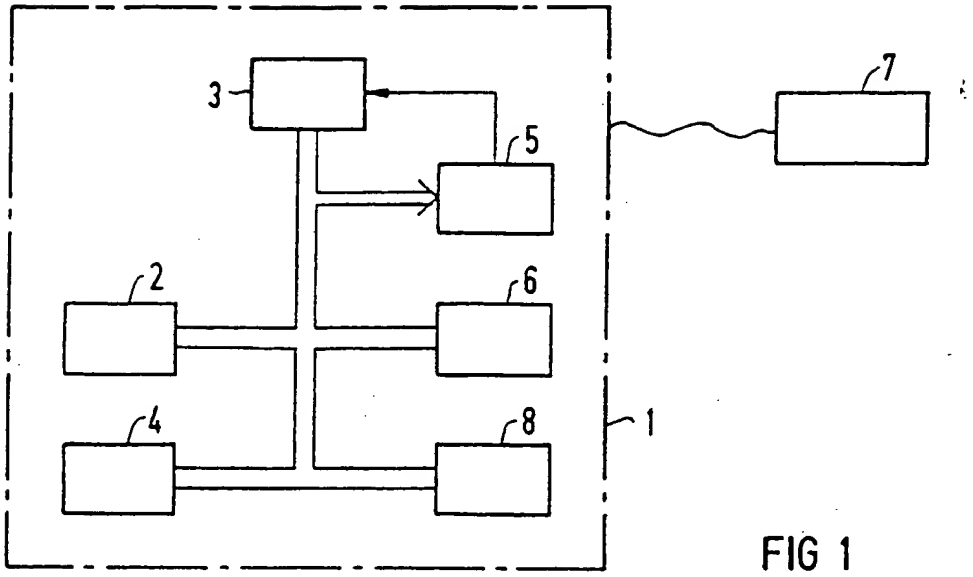

3
7
5
2
6
4
8
1

FIG 1

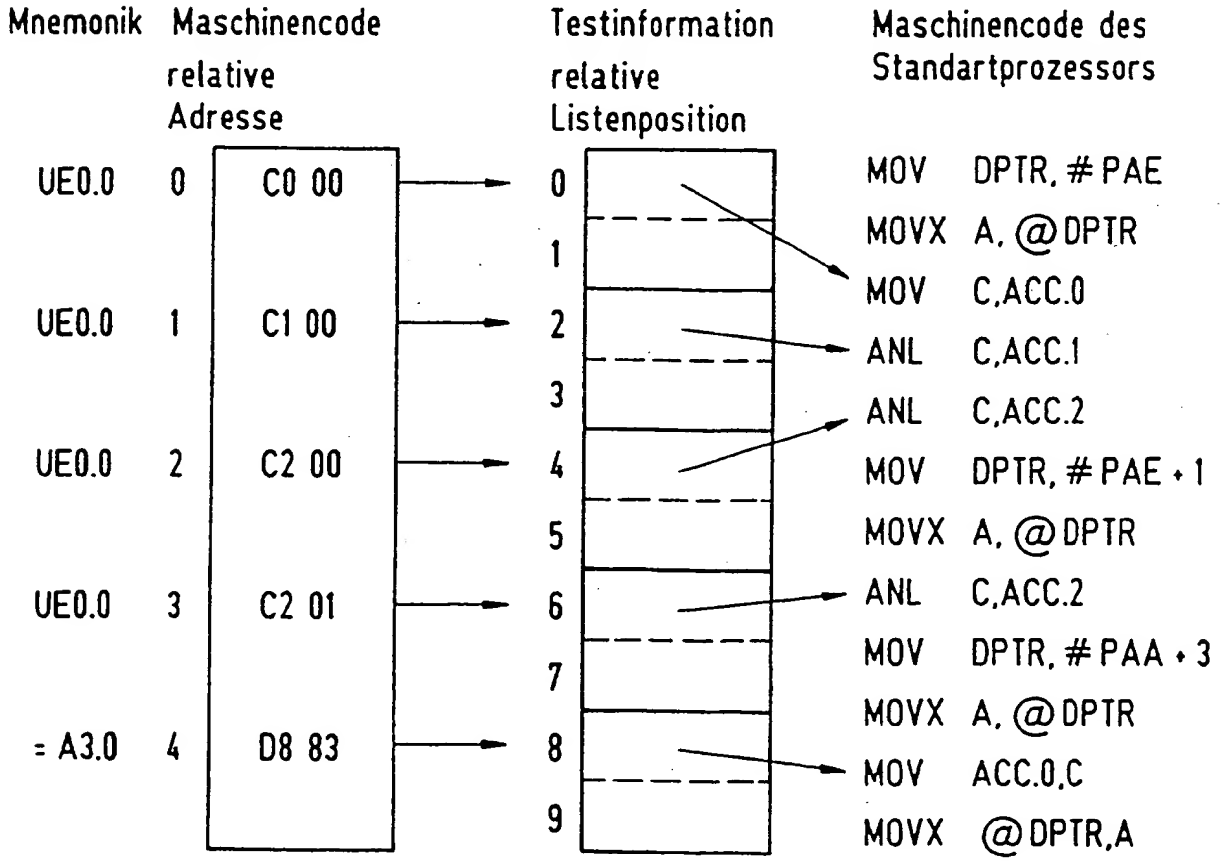
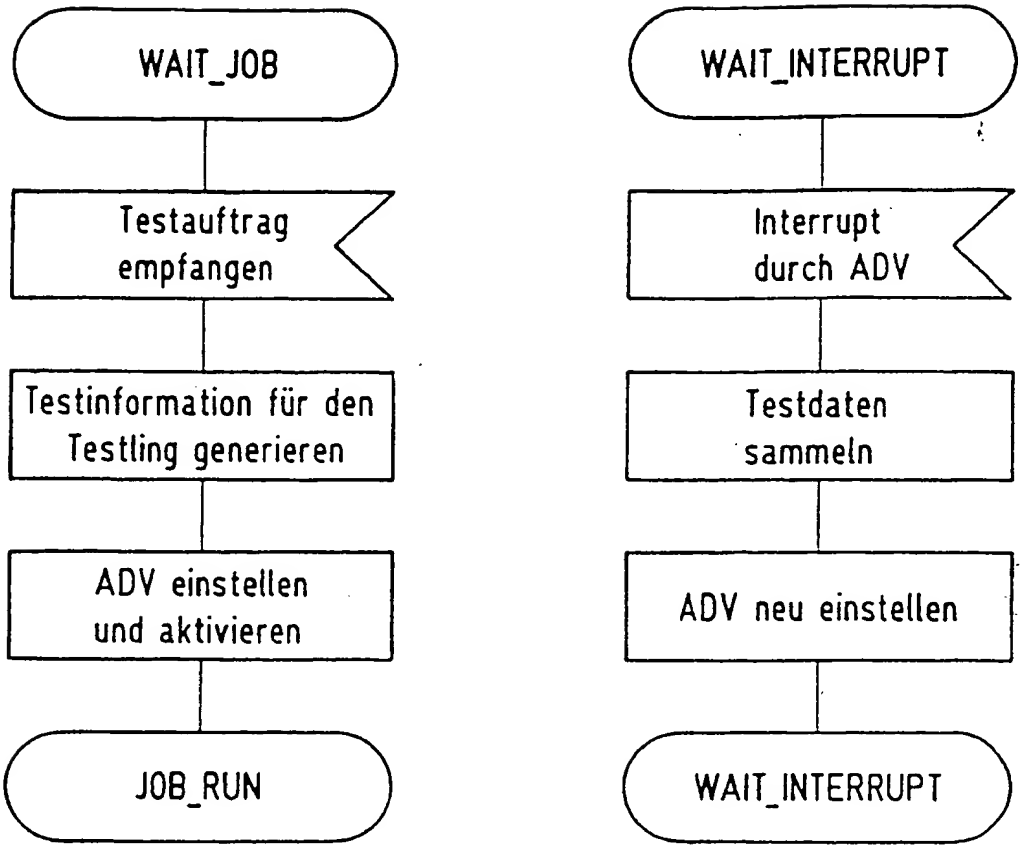
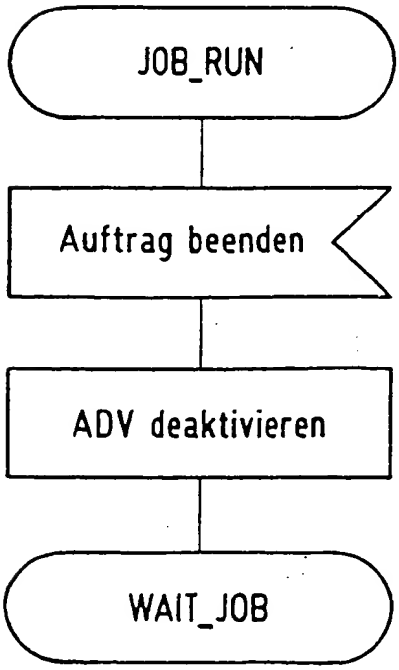

Mnemonik
Maschinencode
relative Adresse
Testinformation
relative Listenposition
Maschinencode des Standartprozessors
UE0.0 0 C0 00
UE0.0 1 C1 00
UE0.0 2 C2 00
UE0.0 3 C2 01
= A3.0 4 D8 83
0 1 2 3 4 5 6 7 8 9
MOV DPTR, # PAE
MOVX A, @DPTR
MOV C,ACC.0
ANL C,ACC.1
ANL C,ACC.2
MOV DPTR, # PAE + 1
MOVX A, @DPTR
ANL C,ACC.2
MOV DPTR, # PAA + 3
MOVX A, @DPTR
MOV ACC.0,C
MOVX @DPTR,A

FIG 2





AG - Protokoll

FIG 3

Europäisches Patentamt

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung

EP 90 10 6297

EINSCHLÄGIGE DOKUMENTE

Kategorie	Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile	Betrifft Anspruch	KLASSIFIKATION DER ANMELDUNG (Int. Cl.5)
Y	PATENT ABSTRACTS OF JAPAN, Band 301, Nr. 415 (P-102), 1990; & JP-A-2 012 344 (MATSUSHITA ELECTRIC) 17-01-1990 * Insgesamt *	1,2	G 06 F 11/00
A	IDEM	5	
Y	FR-A-2 612 661 (REIGA) * Figur 1; Seite 2, Zeile 19 - Seite 3, Zeile 6; Seite 5, Zeile 8 - Seite 6, Zeile 15 *	1,2	
A	IBM TECHNICAL DISCLOSURE BULLETIN, Band 22, Nr. 6, November 1979, Seiten 2578-2583, Armonk, New York, US; H.P. SCHLAEPPI et al.: "Debugging system compatible with optimizing compiler" * Seite 2578, Zeilen 1-32; Seite 2581, Zeilen 10-28 *	1,5	

RECHERCHIERTE SACHGEBIETE (Int. Cl.5)

G 06 F 11
G 06 F 9
G 05 B 19

Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt

Recherchenort	Abschlußdatum der Recherche	Prüfer
Den Haag	23 November 90	GORZEWSKI M.

KATEGORIE DER GENANNTEN DOKUMENTE

X: von besonderer Bedeutung allein betrachtet
Y: von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie
A: technologischer Hintergrund
O: nichtschriftliche Offenbarung
P: Zwischenliteratur
T: der Erfindung zugrunde liegende Theorien oder Grundsätze

E: älteres Patentdokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist
D: in der Anmeldung angeführtes Dokument
L: aus anderen Gründen angeführtes Dokument

&: Mitglied der gleichen Patentfamilie, übereinstimmendes Dokument